WHAT IS CLAIMED IS:

1. A video codec for encoding/decoding digitized sequence of video frames with high compression efficiency, comprising:

- means for frame encoding;
- means for setting and storing codec setting parameters;
- means for controlling desired frame encoding time and CPU loading;
- means for rate control including size of frame encoding output bitstream; and
- means for arithmetic coding of quantized transform coefficients.

2. The video codec of claim 1, comprising means for storing reference frames.

3. The video codec of claim 1, comprising means for performing matching downscaling of a frame before encoding and upscaling of decoded frame according to the coding setting parameters.

4. The video codec of claim 1, deblocking means for processing reconstructed frame texture to eliminate blocking effect from restored data encoded at high distortion level.

5. The video codec of claim 1, further comprising noise suppression means.

6. The video codec of claim 1, comprising means for three-dimensional (3-D) frame encoding/decoding.

7. The video codec of claim 1, comprising means for motion compensation to perform motion estimation, frame head coding, macroblock encoding and coded frame reconstruction and storage.

8. The video codec of claim 7, comprising means for macroblock encoding, comprising:

means for intra prediction,

at least one means for inter prediction,

means for selecting macroblock type and encoding setting,

means for calculating macroblock texture prediction and prediction error,

means for performing texture prediction error transform and transform coefficient quantization;

means calculating motion vector prediction and prediction error; and

means for entropy encoding providing arithmetic context encoding of motion vectors, header parameters and transform coefficients.

9. The video codec of claim 1, further comprising means for selecting a codec mode between motion compensation and 3-D encoding/decoding, depending on desired reconstructed sequence quality and bitrate parameters.

10. The video codec of claim 1, comprising decoding means performing arithmetic context-based decoding using decoding modeling corresponding to arithmetic encoding of the codec.

11. The video codec of claim 10, comprising means for 3-D inverse transform and dequantization.

12. The video codec of claim 10, wherein the decoding means comprises means for motion vector reconstruction, transform coefficient inverse quantization, texture prediction inverse transform, and reconstructed macroblock texture unit.

13. A method of real-time encoding a digitized sequence of video frames using a codec with high compression efficiency, comprising steps of:

- dividing a video frame into macroblocks of pixels;
- performing texture prediction using reconstructed texture of previously encoded/decoded video data;
- performing a texture prediction error transform; and
- performing quantization and encoding of DCT transform coefficients.

14. The method of claim13, comprising a step of downscaling before encoding of the video frame using bilinear interpolation.

15. The method of claim 13, comprising a step of controlling parameters of encoded frames.

16. The method of claim 13, comprising a step of controlling frame encoding time and CPU load.

17. The method of claim 13, comprising a step of selecting best parameters and encoding mode for macroblock coding based on preset coding parameters and codec working parameters.

18. The method of claim 13, comprising a step of noise suppression.

19. The method of claim 17, wherein the encoding mode is a low-complexity 3-dimensional data coding.

20. The method of claim 17, wherein the encoding mode is motion compensation.

21. The method of claim 20, wherein frame encoding starts with choosing a best prediction mode.

22. The method of claim 21, wherein the prediction mode is inter prediction mode predicting block pixels using reconstructed texture of previously coded/decoded frames and specifying block motion vectors.

23. The method of claim 21, wherein the prediction mode is intra prediction mode predicting block pixels using reconstructed texture of previously coded/decoded blocks of current frame and specifying prediction method.

24. The method of claim 23, comprising wavelet transform, wherein resulting wavelet transform coefficients are compressed by context-based entropy coding.

25. The method of claim 24, wherein uniform quantization with constant step size is applied to all wavelet transform coefficients.

26. The method of claim 24, wherein the context- based entropy coding is based on contexts including three neighboring coefficients and one root coefficient, the value of each coefficient being coded arithmetically, and the context-based entropy coding of absolute value of transform coefficients is determined in accordance with the following algorithm:

- set a current value of coefficient = 0;
- construct bits of context for entropy-coded binary value:
  - bit 0 = abs(**n1**) > current value, where abs (n1) is absolute value of the first neighboring coefficient;
  - bit 1 = abs(**n2**) > current value, where abs (n2) is absolute value of the second neighboring coefficient;
  - bit 2 = abs(**n3**) > current value, where abs (n3) is absolute value of the third neighboring coefficient;
  - bit 3 =0 (root coefficient = 0);
  - bits 4,5 = (abs(**n3**) *3 + abs(**n1**)*3 + abs(**n2**)*2 + 4)/8 = {0, 1, 2, 3 or greater};
- using the context, send bit "1" if abs(coefficient) =current value, otherwise send bit "0";
- increment the current value;
- if abs(coefficient) ≠ current value, repeat the construct step;
- if abs(coefficient) > 0, sent a sign,

wherein the bits of context number are: Bit 0 = (n1>0); Bit 1 = (n3>0).

27. The method of claim 25, wherein the uniform quantization of transform coefficients is presented as follows:

q_Coeff = round (**Coeff / Quantizer**),

wherein **Coeff** – wavelet transform coefficient;

q_Coeff – quantized coefficient;

Quantizer – quantization step size.

28. The method of claim 20, comprising step of motion estimation for calculation of components of motion vectors.

29. The method of claim 28, wherein the motion vectors are calculated with quarter-pel accuracy.

30. The method of claim 28, wherein the motion estimation comprises:

- calculating motion vectors *MV(wb, hb, CF, RF)* with integer-pel accuracy using previously calculated motion data;

- calculating motion vectors *MV(wb, hb, CF, RF) [block_x][block_y]* performing inverse logarithmic motion search with parameters *block_x, block_y, current _range*;

- performing motion vector refinement choosing from sets of neighboring motion vectors *MVNeiborhood (wb, hb, CF, RF)[block_x][block_y]* elements *(mvx, mvy)* that provide minimum value of motion vector weight function *Q(mvx, mvy, CF, RF, wb, hb, block_x, block_y)*;

- performing motion vector estimation with quarter-pel accuracy based on results of motion vector estimation with integer-pel accuracy by changing components of the integer-pel accuracy motion vector *MV(wb, hb, CF, RF) [block_x][block_y]* in range *[-3/4; +3/4]* with a step *1/4*; and

- calculating motion vectors *MV(wb, hb, CF, RF)* with quarter-pel accuracy by sequentially applying the motion estimation steps with integer-pel accuracy and with quarter-pel accuracy,

wherein *CF*- current frame with horizontal coordinate *x* and vertical coordinate *y*;

RF- reference frame with horizontal coordinate *x* and vertical coordinate *y*;

wb - width of the blocks for which motion estimation is performed;

hb - height of the blocks for which motion estimation is performed;

W- a multiple of *wb,* current and reference frame width;

H- a multiple of *hb*, current and reference frame height;

Q(mvx, mvy, CF, RF, wb, hb, block_x, block_y) - motion vector weight calculation function;

MV(wb, hb, CF, RF)[block_x][block_y]- motion vector (i.e. pair *(mvx,mvy)* of integers) corresponding to the frame *CF* and reference frame *RF* for a block of width *wb*, height *hb*, which left-top corner is located at a pixel with horizontal coordinate *block_x* and vertical coordinate *block_y*;

MV(wb, hb, CF, RF)- a set of motion vectors *MV(wb, hb, CF, RF)[block_x]* *[block_y]* for: $block\_x = 0, wb, 2 \cdot wb, 3 \cdot wb, \ldots, block\_x < W$, and $block\_y = 0, hb, 2 \cdot hb, 3 \cdot hb, \ldots, block\_y < H$.

MVNeighborhood (wb, hb, CF, RF)[block_x][block_y] - a set of neighboring motion vectors *MV(wb, hb, CF, RF)[nx][ny]*, where *nx* may be equal to $block\_x - wb$, $block\_x$, $block\_x+wb$, and *ny* may be equal to $block\_y - hb$, $block\_y$, $block\_y + hb$, and $nx \geq 0, ny \geq 0, nx \leq W - wb, ny \leq H - hb$.

31. The method of claim 28, comprising step of arithmetic encoding of motion vector prediction difference.

32. The method of claim 13, wherein quantization of DCT transform coefficients is performed using the following formula:

$$q = (c \cdot A\,(Quantstep) + round\_cons) + round\_const) / 2^{20};$$

where c- coefficient value;

q -quantized coefficient value;

A- constant depending on quantization parameter index;

round_const - rounding control: 0.5 *sign (c)*, if $|c| < 20 \cdot 2^{20}/A\,(Quantstep)$ and 0.25 *sign (c)*, if $|c| \geq 20 \cdot 2^{20}/A\,(Quantstep)$.

33. The method of claim 13, wherein encoding of DCT transform coefficients is performed by arithmetic coding based on two-dimensional contest/position-depending modeling.

34. A method of decoding of sequence of video frames encoded according to claim 13, comprising steps of:

- arithmetic decoding;

- decoding coded block pattern of macroblock mode and texture using arithmetic context-based modeling;

- decoding texture prediction error using arithmetic context-based modeling;

- calculating prediction for motion vectors; and

- decoding motion vectors using context-based arithmetic modeling.

35. The method of decoding of claim 34, comprising internal bilinear upscaling correlated with bilinear downscaling provided at the time of encoding.

36. The method of decoding of claim 34, wherein the texture prediction error is provided by inverse transform and dequantization correlated with corresponding encoding procedures.

37. The method of decoding of claim 34, comprising step of deblocking of decoded video frame using at least one of horizontal and vertical deblocking passes for smoothing of sequence of video frame border points.